BỘ GIÁO DỤC
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

TRẦN THỊ THÚY TRINH

KHAI PHÁ TẬP MỤC PHỔ BIẾN MỜ DỰA TRÊN CẤU TRÚC CÂY VÀ KỸ THUẬT XỬ LÝ SONG SONG

Chuyên ngành: Hệ thống thông tin

Mã số: 9 48 01 04

TÓM TẮT LUẬN ÁN TIẾN SĨ NGÀNH MÁY TÍNH

Hà Nội - 2023

Công trình được hoàn thành tại: Học viện Khoa học và Công nghệ - Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Người hướng dẫn khoa học 1: PGS.TS. Nguyễn Long Giang

Người hướng dẫn khoa học 2: TS. Trương Ngọc Châu

Phản biện 1:

Phản biện 2:

Phản biện 3:

Luận án được bảo vệ trước Hội đồng chấm luận án tiến sĩ, họp tại Học viện Khoa học và Công nghệ - Viện Hàn lâm Khoa học và Công nghệ Việt Nam vào hồi….giờ, ngày …tháng … năm 2023.

Có thể tìm hiểu luận án tại:

- Thư viện Học viện Khoa học và Công nghệ
- Thư viện Quốc gia Việt Nam

MỞ ĐẦU

1. Tính cấp thiết của luận án và động lực nghiên cứu

Nghiên cứu gắn với ứng dụng thực tiễn là hoạt động cần nhiều thời gian và công sức không nhỏ của các nhà khoa học. Hơn nữa, trong thời đại công nghệ 4.0, các ứng dụng không chỉ hỗ trợ các tính năng kinh doanh cơ bản mà còn giúp con người đưa ra những dự đoán tương đối chính xác ở thời điểm hiện tại và tương lai. Sự phát triển mạnh mẽ của các hệ thống thông minh này làm tăng nhu cầu ứng dụng thực tế dẫn đến việc tạo ra một lượng lớn dữ liệu hàng ngày. Các công cụ và phương pháp thống kê truyền thống dựa trên nhu cầu ứng dụng, nhưng chúng không có khả năng xử lý lượng dữ liệu khổng lồ có nguồn gốc từ các ứng dụng này. Việc phân tích những dữ liệu như vậy là nhiệm vụ ưu tiên hàng đầu nếu không nó sẽ chuyển sang một hệ thống rất phức tạp và bất lợi. Để khắc phục vấn đề này, khai phá dữ liệu [1]–[3] là một trong những cách tiếp cận có lợi bằng cách hỗ trợ phân tích dữ liệu và tóm tắt dữ liệu thành thông tin hữu ích. Khái niệm khai phá dữ liệu là tạo ra thông tin chưa được xác định trước đó với mức độ liên quan lớn từ cơ sở dữ liệu để ra quyết định. Phụ thuộc vào sự đa dạng của kiến thức, các phương pháp khai phá dữ liệu có thể được chia thành các loại: luật kết hợp [4]–[8], phân loại [7], [9]–[11], phân cụm [12]–[14] và các mẫu tuần tự [15], [16]. Đặc biệt, khai phá luật kết hợp rất quan trọng đối với nghiên cứu khai phá dữ liệu [17]–[19]. Trong các giao dịch kinh doanh phổ biến, luật kết hợp có dạng $A \rightarrow B$ với mục đích tìm kiếm mối quan hệ của các mục trong cơ sở dữ liệu. Điều này giúp doanh nghiệp đưa ra quyết định trong việc hoạch định chiến lược kinh doanh, tiếp thị. Trong giai đoạn thứ nhất của quy trình khai phá luật kết hợp, các tập phổ biến được lấy từ một tập hợp dữ liệu nhất định. Từ các tập mục phổ biến được trích xuất, các luật kết hợp được xây dựng trong giai đoạn thứ hai. Giai đoạn chính của khai phá luật kết hợp là khai phá tập mục phổ biến vì cần rất nhiều nỗ lực để định vị các tập phổ biến trong một tập dữ liệu. Hầu hết các nghiên cứu trong lĩnh vực này đều tập trung vào việc nâng cao hiệu quả khai phá theo nhóm mục phổ biến về mặt thời gian và bộ nhớ.

Các thuật toán khai phá tập mục phổ biến hoặc luật kết hợp truyền thống [20], [21] hầu hết chỉ biểu diễn dữ liệu giao dịch ở dạng giá trị nhị phân, nghĩa là nó liên quan đến sự xuất hiện của các mục; tuy nhiên, với cách tiếp cận rõ, để khai phá các tập mục phổ biến cho các luật kết hợp trong cơ sở dữ liệu có chứa dữ liệu định lượng là khó. Do tính dễ sử dụng và tương tự với suy luận của con người, lý thuyết tập mờ [22], [23] đang được sử dụng trong các hệ thống thông minh thường xuyên hơn [24]–[27]. Biểu diễn ngôn ngữ làm cho tri thức đơn giản hơn để con người dễ hiểu, do đó nó được sử dụng rộng rãi. Vì vậy, để khai phá các luật kết hợp mờ từ cơ sở dữ liệu định lượng, các miền của thuộc tính định lượng sẽ được chuyển đổi thành một tập mờ được thể hiện trong các biến ngôn ngữ bằng cách sử dụng hàm liên thuộc [28], cách tiếp cận này có thể làm giảm các tính toán. Một số thuật toán khai phá mờ đã được nghiên cứu và phát triển rộng rãi sử dụng lý thuyết tập mờ để chuyển đổi giá trị định lượng của mục thành các thuật ngữ ngôn ngữ dựa trên cơ chế giống như Apriori thông thường [29], [30], [31] [32].

Thuật toán cho khai phá luật kết hợp mờ với hiệu suất nhanh và hiệu quả trên các tập dữ liệu lớn được đề xuất bởi Mangalampalli và Pudi [33]. Tác giả đã sử dụng phương pháp tidlist để tính tần suất của vị trí đặt, với các tính năng như cấu trúc dữ liệu sử dụng byte-vector biểu diễn tidlist, danh sách nén đã sử dụng góp phần tăng hiệu suất. Trước đây, Janikow đã kết hợp các cây quyết định tượng trưng trên các hệ thống dựa trên luật để điều khiển mờ [34] bằng cách sử dụng biểu diễn mờ. Watanabe và Fujioka [35], [36] đã định nghĩa sự dư thừa tương đương của các phần tử mờ và các định lý liên quan cho việc khai phá luật kết hợp mờ. Mục tiêu của thuật toán là tinh chỉnh thời gian dành cho việc khai phá luật và đồng thời cắt bỏ các luật thừa trong các ứng dụng khai phá dữ liệu. Tuy nhiên, hầu hết các phương pháp khai phá luật kết hợp mờ áp dụng Apriori [37] để

tạo ra các ứng cử viên và kiểm tra sự hỗ trợ của chúng, do đó yêu cầu quét lại cơ sở dữ liệu nhiều lần, vì vậy nó gây ra quá trình chậm và không hiệu quả trong cơ sở dữ liệu lớn. Hơn nữa, với cách biểu diễn mờ trong các thuật toán trên, tập hợp mờ của các thuộc tính định lượng và hàm thành viên của chúng phụ thuộc vào ý kiến chủ quan của chuyên gia hoặc tính sẵn có. Vấn đề này gây ra ranh giới "sắc nét" giữa các khoảng mờ, vì vậy khó có thể xác định mức độ của hàm liên thuộc cho các phần tử gần ranh giới của khoảng. Đây là khoảng trống thứ nhất được xác định trong vấn đề nghiên cứu của luận án.

Thay vì sử dụng cách tiếp cận thông thường theo Apriori, Lin et al. đã triển khai phương pháp cây phổ biến mờ (FFP)-tree [38], [39] để khai phá tập mục phổ biến mờ dựa trên cơ chế phát triển mẫu. Tiếp cận này đã áp dụng cả lý thuyết tập mờ và cấu trúc cây FP (Frequent pattern) để xây dựng cây FFP (Fuzzy Frequent Pattern) có thể được sử dụng cho quá trình khai phá. Các biến ngôn ngữ được chuyển đổi cùng với mức độ thuộc của chúng được sắp xếp theo thứ tự tăng dần của mỗi giao dịch, do đó giữ được tính chất đóng (downward closure property) để xây dựng đệ quy cây điều kiện và khai phá các mục phổ biến mờ cần thiết. Cách tiếp cận này có thể yêu cầu rất nhiều thời gian tính toán khi quy mô giao dịch rất lớn. Thuật toán nén cây phổ biến mờ (CFFP – Compact Fuzzy Frequent Pattern)-tree [40] sau đó được thiết kế để giảm kích thước của cây FFP. Do đó, một mảng được gắn với mỗi nút bằng cách bảo toàn các giá trị mờ cho biến ngôn ngữ được xử lý hiện tại với bất kỳ tập mục tiền tố nào của nó trong đường đi. Mặc dù số lượng nút cây của cây CFFP giảm đáng kể so với thuật toán cây FFP, nhưng cần phải giữ thêm một mảng của mỗi nút để lưu trữ các giá trị thành viên của nút được xử lý hiện tại với bất kỳ biến ngôn ngữ nào của nó trong đường đi. Do đó, nó yêu cầu dung lượng bộ nhớ để lưu giữ những thông tin đó, điều này không hiệu quả trong một cơ sở dữ liệu thưa. Để giải quyết hạn chế này, thuật toán cây mẫu phổ biến mờ giới hạn trên (UBFFPT - upper-bound fuzzy frequent pattern) [41] sau đó được thiết kế để giữ không chỉ cấu trúc cây dày đặc mà còn có thể khai phá các tập mục phổ biến mờ từ giới hạn bộ nhớ so với cây FFP và thuật toán cây CFFP. Thuật toán cây UBFFPT có thể khai thác hiệu quả các mục phổ biến mờ giữ nguyên kích thước của các nút cây như thuật toán cây CFFP nhưng việc sử dụng bộ nhớ và tính toán có thể giảm đáng kể. Các thuật toán trên chỉ sử dụng một thuật ngữ ngôn ngữ duy nhất để biểu diễn mục được xử lý trong cơ sở dữ liệu, do đó thông tin được phát hiện có thể không đầy đủ. Nhiều thuật toán liên quan đến khai phá tập phổ biến mờ kép [42]–[44] được đề xuất nhằm giúp tri thức được khai phá đầy đủ hơn so với các phương pháp truyền thống. Hong và cộng sự [42] sau đó đã phát triển cấu trúc dựa trên cây với ý tưởng tương tự về cây FP và FFPT [38] nhưng duy trì nhiều tập mục phổ biến mờ 1-item với cây MFFP được thiết kế để khai phá thông tin cần thiết, không chỉ biến ngôn ngữ đơn lẻ được giữ để biểu diễn cho một mục mà tất cả các mục có giá trị mờ của chúng không nhỏ hơn ngưỡng hỗ trợ tối thiểu. Vì vậy, một thông tin đầy đủ hơn được lưu giữ để ra quyết định hiệu quả. Hơn nữa, ý tưởng tương tự sau đó được áp dụng cho cây CMFFP [43] và cây UBMFFP [44]. Với thông tin đầy đủ hơn về nhiều mẫu phổ biến mờ dẫn xuất, các chiến lược hiệu quả do đó có thể đạt được để ra quyết định. Tuy nhiên, trong các thuật toán này, việc khai phá các tập phổ biến mờ được thực hiện một cách đệ quy từ cấu trúc cây, do đó nó yêu cầu một bộ nhớ lớn để lưu trữ các cây tạm thời. Đây là khoảng trống thứ hai luận án sẽ giải quyết.

Khai phá tập phổ biến từ nhiều tập dữ liệu mờ được đề cập trong bài báo [45]. Trong bài báo, tác giả kết hợp nhiều bảng bằng cách sử dụng lược đồ sao tìm các luật kết hợp đa cấp mờ trong mô hình cơ sở dữ liệu quan hệ, có khả năng xử lý nhiều bảng. Thuật toán sử dụng phép nối và thực thể để nhận ra các tập mục phổ biến. Tuy nhiên, kết quả của bài báo vẫn còn nhiều hạn chế trong việc tính toán hỗ trợ của các tập mục liên quan

đến các kết nối khác có chứa thuộc tính mờ. Phương pháp khác như [46] sử dụng thuật toán tiến hóa vi phân (DE) để khai phá các luật kết hợp mờ có ý nghĩa thống kê được tối ưu hóa có số lượng lớn và các giá trị đo lường có nghĩa với sự kiểm soát chặt chẽ đối với rủi ro của các luật suy đoán. Ngoài ra, thuật toán dựa trên mẫu được đề xuất trong [47] nhằm mục đích tìm các luật kết hợp mờ từ các tập dữ liệu định lượng lớn. Nhiều nghiên cứu khác nhau đã được thực hiện không chỉ để cải thiện hiệu suất mà còn cải thiện tốc độ tìm kiếm các luật kết hợp mờ với bảng băm, lược đồ hoặc cấu trúc dữ liệu cây [40], [41], [43], [44]. Thuật toán khai phá tập mục mờ phổ biến FFI-Miner [48] được phát triển để khai phá tập đầy đủ các FFI mà không cần tạo ứng viên. Thuật toán sử dụng chiến lược cắt tỉa hiệu quả cũng được phát triển để giảm không gian tìm kiếm, do đó đẩy nhanh quá trình khai phá để phát hiện trực tiếp các tập mục mờ phổ biến. Các mẫu phổ biến là các tập mục được tìm thấy trong một số lượng đáng kể các giao dịch. Cùng với sự gia tăng kích thước dữ liệu, các loại dữ liệu không đồng nhất và biến thể dữ liệu cực kỳ động. Do đó, việc mở rộng các thuật toán khai phá mờ hiệu quả cho kỷ nguyên dữ liệu lớn là một vấn đề quan trọng việc khai phá bằng cách áp dụng các kỹ thuật xử lý song song đã trở thành một cách khả thi để khắc phục vấn đề thời gian xử lý. Đây là khoảng trống thứ ba được xác định trong luận án.

Tại Việt Nam, khai phá luật kết hợp đã được các nhóm nghiên cứu tại Viện Công nghệ Thông tin thuộc Viện Khoa học và Công nghệ Việt Nam như luận án tiến sĩ của Nguyễn Huy Đức [49] giới thiệu thuật toán FSM là thuật toán nhanh khai phá tất cả các tập mục cổ phần cao trong cơ sở dữ liệu giao tác và đề xuất thuật toán AFSM (Advanced FSM) dựa trên các bước của thuật toán FSM với phương pháp mới tỉa hiệu quả hơn các tập mục ứng viên. Luận án tiến sĩ của Nguyễn Long Giang [50] trình bày về phương pháp khai phá dữ liệu sử dụng lý thuyết tập thô. Bài báo của tác giả Nguyễn Công Hào [51] trình bày một phương pháp xử lý luật kết hợp mờ dựa trên đại số gia tử. Nhóm nghiên cứu của PGS. TS. Võ Đình Bảy và GS. TS. Lê Hoài Bắc đưa ra phương pháp khai phá tập mục phổ biến trong cơ sở dữ liệu rõ như [52]–[55], đây có thể được xem là nền tảng cho những nghiên cứu trong luận án.

Luận án này nhằm giải quyết ba khoảng trống được xác định ở trên. Việc nghiên cứu giải quyết những vấn đề đó là thực sự cần thiết không chỉ ở phương diện phát triển lý thuyết mà cả ở phương diện ứng dụng thực tế. Đó là động lực để tác giả luận án thực hiện nghiên cứu đề tài "**Khai phá tập mục phổ biến mờ dựa trên cấu trúc cây và kỹ thuật xử lý song song**" để đưa ra các phương pháp mới hiệu quả về khai phá tập mục phổ biến và khai phá các luật kết mờ dựa trên lý thuyết tập mờ.

2. Mục tiêu, đối tượng và phạm vi nghiên cứu của luận án

a. Mục tiêu nghiên cứu

Mục tiêu của luận án nhằm đề xuất các giải pháp khai phá tập mục phổ biến mờ trong cơ sở dữ liệu định lượng, khắc phục vấn đề "sắc nét" khi phân vùng dữ liệu mờ cho các thuộc tính có giá trị định lượng.

Cụ thể, luận án tập trung đề xuất các giải pháp nhằm:

- Xác định các tập mờ cho mỗi thuộc tính định lượng trong cơ sở dữ liệu thông qua kỹ thuật phân cụm.
- Giảm bộ nhớ lưu trữ trong quá trình khai phá tập mục phổ biến mờ
- Giảm thời gian xử lý trong việc khai phá các tập mục phổ biến mờ trong các cơ sở dữ liệu lớn.

b. Đối tượng nghiên cứu

- Các thuật toán khai phá tập mục phổ biến trong cơ sở dữ liệu giao dịch
- Các thuật toán khai phá tập mục phổ biến mờ, khai phá luật kết hợp mờ trong cơ sở dữ liệu định lượng.

c. Phạm vi nghiên cứu

- Luận án nghiên cứu các luật kết hợp mờ, tập mục phổ biến mờ trong cơ sở dữ liệu định lượng.
- Tổng hợp các công bố khoa học liên quan đến các phương pháp khai phá tập mục phổ biến mờ.
- So sánh thực nghiệm với các thuật toán đã có

3. Phương pháp nghiên cứu

Luận án đã sử dụng các phương pháp nghiên cứu sau:

- Tổng hợp và đánh giá các kết quả đã được công bố về các phương pháp khai phá tập mục phổ biến mờ từ nhiều nguồn thông tin thu thập được. Trên cơ sở đó đề xuất các kết quả mới, đánh giá kết quả mới bằng việc cài đặt thử nghiệm một số thuật toán. Áp dụng kết quả để giải quyết một bài toán trong thực tiễn.
- Phương pháp so sánh: được sử dụng để so sánh các kỹ thuật, thuật toán đã được đề xuất để giải quyết những vấn đề nghiên cứu liên quan, từ đó hình thành ý tưởng cho thuật toán mới cho vấn đề nghiên cứu.
- Phương pháp thực nghiệm: Các thuật toán được đề xuất đều được thực nghiệm trên các tập dữ liệu thực để đánh giá sự đúng đắn và tính khả thi của thuật toán.

4. Các đóng góp chính của luận án

Những đóng góp chính của luận án là đề xuất và giải quyết các vấn đề sau:

- Đề xuất phương pháp xác định các tập mờ cho mỗi thuộc tính định lượng trong cơ sở dữ liệu thông qua kỹ thuật phân cụm. Cụ thể hơn, luận án trình bày kỹ thuật phân cụm EMC. Mục tiêu của các thuật toán này là chia dữ liệu thành các cụm có ý nghĩa. Sau đó, các cụm này được sử dụng để phân loại mỗi thuộc tính định lượng như một tập mờ và xác định các hàm thuộc của chúng. [CT2], [CT4].
- Đề xuất phương pháp khai phá tập mục phổ biến mờ trong cơ sở dữ liệu định lượng sử dụng cấu trúc dữ liệu Node-list. Quy trình khai phá tập mục mờ phổ biến dựa trên PP_code hoặc POS_code giúp hạn chế mức tiêu thụ bộ nhớ được yêu cầu. [CT1], [CT2], [CT5]
- Đề xuất một phương pháp xử lý song song để khai phá các tập mờ phổ biến sử dụng phương pháp tiếp cận automata di động học(Cellular learning automata). Theo CLA, không gian được biểu diễn như một mạng, với mỗi phần tử là một ô. Từng dòng một, dữ liệu giao dịch sẽ được đọc và đồng thời được chuyển đến các ô, chúng xử lý song song với nhau. Thông qua việc sử dụng các ô dữ liệu tự trị này, việc khai phá các tập mục mờ phổ biến được thực hiện. Quá trình này rút ngắn thời gian thực thi của thuật toán. [CT3].

5. Bố cục luận án

Luận án gồm phần Mở đầu, 03 chương và phần kết luận.

- Phần Mở đầu: Trình bày sự cần thiết và động lực nghiên cứu của đề tài; mục tiêu, đối tượng, phạm vi nghiên cứu; phương pháp nghiên cứu; những đóng góp chính và cấu trúc của luận án.
- Chương 1: Cơ sở lý thuyết
- Chương 2: Các phương pháp khai phá tập mục phổ biến mờ dựa trên cấu trúc cây.
- Chương 3: Khai phá tập mục phổ biến mờ sử dụng phương pháp xử lý song song.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

Trong chương này, NCS trình bày các khái niệm cơ bản về luật kết hợp, luật kết hợp định lượng, logic mờ, luật kết hợp mờ và các nghiên cứu liên quan đến luật kết hợp mờ. Từ đó, xác định các vấn đề còn tồn tại cần giải quyết trong chương 2.

1.1 Luật kết hợp

1.1.1 Các khái niệm cơ bản về luật kết hợp [55]

Định nghĩa 1.1 Cơ sở dữ liệu giao tác:

Giả sử $I = \{i_1, i_2, \dots, i_m\}$ là tập các mục. $D = \{T_1, T_2, \dots, T_n\}$ là một tập các giao tác, được gọi là cơ sở dữ liệu giao tác, trong đó mỗi giao tác t trong D có dạng (tid, X) trong đó, mỗi giao tác t có định danh tid và tập mục t-itemset, $t = (tid, t - itemset)$; X được gọi là tập mục itemset nếu $X \subseteq I$.

Định nghĩa 1.2: Độ hỗ trợ của tập mục

Độ hỗ trợ của một tập mục *X* trong cơ sở dữ liệu giao tác *D* ký hiệu là *sup (X)* là số giao dịch chứa tập mục X, được tính bởi công thức sau:

$$sup(X) = |t|\ X \subseteq t, t \in D| \tag{1.1}$$

Trong đó ký hiệu |.| là số giao tác.

Định nghĩa 1.3: Tập mục phổ biến

Một tập mục X có trong cơ sở dữ liệu giao tác D được gọi là phổ biến nếu độ hỗ trợ của nó ($sup(X)$) lớn hơn hoặc bằng ngưỡng độ hỗ trợ tối thiểu (minsup) cho trước do người dùng định nghĩa. Vì vậy, độ hỗ trợ được xem là tần suất xuất hiện đồng thời của các mục.

Định nghĩa 1.4: Luật kết hợp

Một luật kết hợp là một mệnh đề kéo theo có dạng X →Y, trong đó X và Y là các tập mục thoả mãn điều kiện: $X \subseteq I$, $Y \subseteq I$ và $X \cap Y = \emptyset$. Đối với luật kết hợp X → Y, X được gọi là tiền đề, Y được gọi là kết quả của luật.

Định nghĩa 1.5 : Độ hỗ trợ của một luật

Cho luật kết hợp $r = X \to Y$, độ hỗ trợ của luật r ký hiệu là sup(r) là tỉ số giữa số lượng các giao tác T ⊆ D có chứa cả tập mục X và tập mục Y với tổng số giao tác trong D được xác định như sau:

$$sup(r) = \frac{|\{T \in D | T \supset X \cup Y\}|}{|D|} \tag{1.2}$$

Định nghĩa 1.6 Độ tin cậy của một luật

Cho luật kết hợp $r = X \to Y$, độ tin cậy của luật r ký hiệu là conf(r) là tỉ số giữa số lượng các giao tác T ⊆ D có chứa cả tập mục X và tập mục Y với tổng số giao tác trong D chứa tập mục X, được xác định như sau:

$$conf(r) = \frac{|\{T \in D | T \supset X \cup Y\}|}{|\{T \in D | T \supset X\}|} = \frac{sup(X \cup Y)}{sup(X)} \tag{1.3}$$

Định nghĩa 1.7: Luật kết hợp mạnh

Cho luật kết hợp $r = X \to Y$, nếu luật r thỏa mãn cả hai ngưỡng là độ hỗ trợ tối thiểu (minsup) và độ tin cậy tối thiểu (minconf) được gọi là luật kết hợp mạnh, tức là:

$$sup(r = X \to Y) = P(X \cup Y) \geq minsup$$

$$conf(r = X \to Y) = P(X \cup Y) = \frac{sup(X \cup Y)}{sup(X)} \geq minconf$$

Phát biểu bài toán: Bài toán luật kết hợp được phát biểu như sau [49]:

Cho một cơ sở dữ liệu giao tác D, độ hỗ trợ tối thiểu minsup, độ tin cậy tối thiểu minconf. Hãy tìm tất cả các luật kết hợp có dạng $X \to Y$ thỏa mãn độ hỗ trợ $sup(X \cup Y) \geq minsup$ và độ tin cậy $conf(X \to Y) = \frac{sup(X \cup Y)}{sup(X)} \geq minconf$

1.1.2 Luật kết hợp trong cơ sở dữ liệu nhị phân

Luật kết hợp nhị phân đề cập đến các luật cổ điển trong bài toán phân tích giỏ hàng. Ở đây các sản phẩm có thể có trong giao dịch hoặc không, chỉ tạo ra các giá trị kiểu boolean (được biểu diễn bằng 1 và 0). Do đó, mọi mục trong giao dịch có thể được xác định là một thuộc tính nhị phân với miền $\{0,1\}$. Mô hình được định nghĩa trong [55] như sau:

Cho $I = \{i_1, i_2, \dots, i_m\}$ là một tập các thuộc tính nhị phân, gọi là các mục. Cho T là cơ sở dữ liệu giao dịch. Mỗi giao dịch t được biểu diễn như là vecto nhị phân với $t[k] = 1$ nếu giao dịch t có chứa mục i_k và $t[k] = 0$ nếu ngược lại. Cho X là một tập mục chứa trong I, ta nói một giao dịch t thỏa mãn X nếu mọi mục trong X $i_k \in X, t[k] = 1$.

1.1.3 Luật kết hợp trong cơ sở dữ liệu định lượng

Theo dạng luật kết hợp nhị phân này thì các mục chỉ được quan tâm là có hay không xuất hiện trong cơ sở dữ liệu giao tác chứ không quan tâm về mức độ hay tần xuất xuất hiện. Trong thực tế, cơ sở dữ liệu không chỉ chứa các thuộc tính nhị phân mà còn chứa các thuộc tính định lượng và phân loại mà không thể khai phá bằng kỹ thuật cổ điển. Việc khai phá các luật trong loại dữ liệu như vậy có thể được gọi là bài toán luật kết hợp định lượng [29]. Chiến lược khai phá luật kết hợp định lượng được thực hiện bằng cách chuyển đổi các thuộc tính có giá trị định lượng sang giá trị nhị phân. Trong phương pháp này, mỗi giá trị định lượng/phân loại có dạng $\langle attribute, value \rangle$ được ánh xạ sang giá trị nhị phân. Sau đó, các kỹ thuật khai phá luật kết hợp nhị phân được thực hiện để tìm luật. Trong khai phá luật kết hợp định lượng, các thuộc tính có thể là định lượng và phân loại.

1.2 Tổng quan về Logic mờ

1.2.1 Tập mờ

Cho một tập vũ trụ U với các phần tử ký hiệu bởi u , $U = \{x\}$. Một tập mờ $\tilde{A}$ trên U là tập được đặc trưng bởi một hàm $\mu_A(u)$ mà nó liên kết mỗi phần tử $u \in U$ với một số thực trong đoạn [0,1].

$$\tilde{A} = \{(u, \mu_A(u)) \mid u \in U\} \quad (1.4)$$

Trong đó $\mu_A(u)$ là một ánh xạ từ U vào [0,1] và được gọi là hàm thành viên của tập mờ $\tilde{A}$.

1.2.2 Hàm thành viên

Hàm thành viên $\mu_A(u)$ định nghĩa cho tập A trên tập vũ trụ U trong khái niệm tập hợp kinh điển chỉ có hai giá trị là 1 nếu $u \in A$ hoặc 0 nếu $u \notin A$. Tuy nhiên trong khái niệm tập mờ thì giá trị hàm thành viên chỉ mức độ thuộc về (membership degree) của phần tử u vào tập mờ A. Khoảng xác định của hàm $\mu_A(u)$ là đoạn [0, 1], trong đó giá trị 0 chỉ mức độ không thuộc về, còn giá trị 1 chỉ mức độ thuộc về hoàn toàn.

$$\mu(A) : U \rightarrow [0,1] \quad (1.5)$$

Kiểu của tập mờ phụ thuộc vào các kiểu hàm thành viên khác nhau. Có nhiều kiểu hàm thành viên khác nhau được đề xuất.

1.2.3 Biến ngôn ngữ

Biến ngôn ngữ [66] là bộ năm $(X, T(X), U, R, M)$, trong đó X là tên biến, T(X) là tập giá trị ngôn ngữ của biến X, U là không gian tham chiếu của biến cơ sở u, mỗi giá trị ngôn ngữ được xem là một biến mờ trên U kết hợp với biến cơ sở u, R là một quy tắc cú pháp sinh các giá trị ngôn ngữ của $T(X)$, M là quy tắc ngữ nghĩa gán mỗi giá trị ngôn ngữ trong $T(X)$ với một tập mờ trên U.

Ví dụ: Cho X là biến ngôn ngữ có tên TUỔI, biến cơ sở u lấy theo số tuổi của con người có miền xác định là $U = [0,100]$. Tập các giá trị ngôn ngữ $T(TUỔI) = \{rất\ trẻ, trẻ, trung\ niên, già, rất già\}$.

1.2.4 Các phép toán logic mờ

Ba phép toán logic mờ cơ bản: phép bù, phép hợp và phép giao thường được sử dụng trong lý thuyết tập mờ, được mô tả dưới đây [22].

Phép bù: Phép toán bù của tập mờ A được ký hiệu là ⌐A. Hàm thành viên của ⌐A có thể được định nghĩa là:

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \quad \forall x \in X \tag{1.9}$$

Phép hợp: Phép hợp của hai tập mờ A và B được ký hiệu là $A \cup B$. Hàm thuộc của $A \cup B$ đối với phép toán chuẩn có thể được định nghĩa như sau:

$$\mu_{A\cup B}(x) = max\{\mu_A(x), \mu_B(x)\}, \qquad \forall x \in X \tag{1.10}$$

Phép giao: phép toán giao của hai tập mờ A và B được ký hiệu là $A \cap B$. Hàm thành viên của $A \cap B$ đối với phép toán chuẩn có thể được định nghĩa như sau:

$$\mu_{A\cap B}(x) = min\{\mu_A(x), \mu_B(x)\}, \qquad \forall x \in X \tag{1.11}$$

1.3 Luật kết hợp mờ

1.3.1 Cơ sở dữ liệu giao dịch mờ

Cho $I = \{I_1, I_2, \dots, I_m\}$ là tập n các thuộc tính, i_u là thuộc tính thứ u trong I. $D_Q = \{T_1, T_2, \dots, T_n\}$ là một tập các giao tác với mỗi $T_v \in D_Q$ là một tập con của I chứa các mục có giá trị định lượng và có một định danh duy nhất TID. Một giao tác T được gọi là chứa X nếu $X \subseteq T_q$ trong đó, X là một tập chứa vài mục có trong I. Mỗi thuộc tính I_k có thể kết hợp với tập giá trị mờ được biểu diễn $F_{ik} = \{f_{ik}^1, f_{ik}^2, \dots, f_{ik}^h\}$ với f_{ik}^jlà giá trị mờ thứ j trong F_{ik}. Sử dụng hàm thành viên liên quan để xác định tập mờ cho các mỗi thuộc tính, cơ sở dữ liệu định lượng D_Q được chuyển thành cơ sở dữ liệu chứa giá trị mờ D_f.

1.3.2 Độ hỗ trợ của tập mục mờ

Một tập thuộc tính mờ trong luật kết hợp mờ là một cặp $\langle X, A\rangle$ với A là tập các tập mờ tương ứng với các thuộc tính trong X và $X \subseteq I$.

Độ hỗ trợ của tập mục $\langle X, A\rangle$ ký hiệu là $fsup(\langle X, A\rangle)$ được xác định bởi công thức sau:

$$fsup(\langle X, A\rangle) = \sum_{t\in T} \mu_{x1}(t) \otimes \mu_{x2}(t) \otimes \dots \otimes \mu_{xp}(t) \tag{1.12}$$

Trong đó, $\mu_{xp}(t)$ là giá trị mờ của thuộc tính x_p trong giao tác *t*.

$\otimes$ là toán từ T-norm (T-chuẩn). Trong lý thuyết logic mờ, nó có vai trò giống như phép toán AND trong logic cổ điển. Có nhiều cách lựa chọn phép toán T-norm như:

Phép lấy min: $a \otimes b = min(a, b)$

Tích đại số: $a \otimes b = ab$

Tích bị chặn: $a \otimes b = max(0, a + b - 1)$

Tích Drastic: $a \otimes b = \begin{cases} a\ (nếu\ b = 1) \\ b\ (nếu\ a = 1) \\ 0\ (nếu\ a, b < 1) \end{cases}$

Phép giao: $a \otimes b = 1 - min\left[1, ((1-a)^w + (1-b)^w)^{\frac{1}{w}}\right]$ với $(w > 0)$

Phép lấy min và phép tính đại số là hai phép toán phù hợp nhất vì nó thuận tiện cho việc tính toán và thể hiện được mối liên hệ chặt chẽ giữa các thuộc tính trong các tập phổ biến.

Khi chọn phép lấy min cho toán tử T-norm, công thức tính độ hỗ trợ của tập mục $\langle X, A\rangle$ trở thành:

$$fsup(\langle X, A\rangle) = \sum_{t\in T} min\{\mu_{x1}(t),\ \mu_{x2}(t), \dots,\ \mu_{xp}(t)\} \tag{1.13}$$

Khi chọn phép lấy tích đại số cho toán tử T-norm, công thức tính độ hỗ trợ của tập

mục $\langle X, A\rangle$ trở thành:

$$sup(\langle X, A\rangle) = \sum_{t\in T}\prod_{x_p\in X}\{\mu_{xp}(t)\} \quad (1.14)$$

1.3.3 Tập mục mờ phổ biến

Định nghĩa 1.8: (Tập mờ phổ biến): [41]

Một tập mục $\langle X, A\rangle$ được gọi là phổ biến nếu độ hỗ trợ của nó lớn hơn hoặc bằng độ hỗ trợ tối thiểu (*fminsup*) do người dùng định nghĩa $fsup(\langle X, A\rangle) \geq fminsup$.

Khai phá tập mục mờ phổ biến là bài toán trích xuất tất cả các tập mục mờ phổ biến có dạng:

$$FFI_k = \{X \mid fsup(X) \geq \delta \times |D_f|\} \quad (1.15)$$

1.3.4 Luật kết hợp mờ

Sau khi có được các khoảng mờ và các hàm thành viên tương ứng của chúng cho mỗi tập mờ của thuộc tính định lượng được, một cơ sở dữ liệu D_F được biến đổi (bằng cách mờ hóa) được tạo ra từ cơ sở dữ liệu gốc. Cho cơ sở dữ liệu mờ $D_F = \{T_1, T_2, \dots, T_n\}$ với các thuộc tính $i_j \in I$ và các tập mờ F_{ij} tương ứng với các thuộc tính trong I. Một luật kết hợp mờ có dạng như sau:

$If\ X = \{x_1, x_2 \dots, x_p\}\ is\ A = \{a_1, a_2 \dots, a_p\}\ then\ Y = \{y_1, y_2 \dots, y_q\}\ is\ B = \{b_1, b_2 \dots, b_q\}$

Trong đó: $a_i \in F(x_i)$, $i = 1, \dots, p$ và $b_j \in F(y_j)$, $j = 1, \dots, q$. X và Y là các tập mục con có trật tự của I và phân biệt, không có chung thuộc tính nào. Khi đó, X is A được gọi là tiền đề của luật và Y is B được gọi là hệ quả của luật.

Một ví dụ về luật kết hợp có dạng: *Nếu Tuổi is Trẻ THEN Thu nhập is Thấp*

Định nghĩa 1.9: (Độ hỗ trợ của một luật kết hợp mờ)

Độ hỗ trợ của một luật mờ $X\ is\ A \Rightarrow Y\ is\ B$ được xác định theo công thức sau:

$$fsup(\langle X\ is\ A \Longrightarrow Y\ is\ B\rangle) = fsup(\langle X \cup Y, A \cup B\rangle) \quad (1.16)$$

Định nghĩa 1.10: (Độ tin cậy của một luật kết hợp mờ)

Độ tin cậy của một luật mờ $X\ is\ A \Rightarrow Y\ is\ B$ được xác định theo công thức sau:

$$fconf(\langle X\ is\ A \Longrightarrow Y\ is\ B\rangle) = \frac{fsup(\langle X\ is\ A \Longrightarrow Y\ is\ B\rangle)}{fsup(\langle X, A\rangle)} \quad (1.17)$$

Định nghĩa 1.11: (Luật mờ phổ biến).

Một luật được gọi là phổ biến nếu độ hỗ trợ của nó lớn hơn hoặc bằng *fminsup*, có nghĩa là $fsup(\langle X\ is\ A \Longrightarrow Y\ is\ B\rangle) \geq fminsup$.

Định nghĩa 1.12. (Luật mờ tin cậy). Một luật được xem là tin cậy nếu độ tin cậy của nó lớn hơn hoặc bằng độ tin cậy tối thiểu fminconf (fuzzy minimum confidence) được định nghĩa bởi người dùng, nghĩa là $fconf(\langle X\ is\ A \Longrightarrow Y\ is\ B\rangle) \geq fminconf$.

1.4 Các nghiên cứu liên quan

1.4.1 Các nghiên cứu tiếp cận dựa trên Apriori

Các nghiên cứu tiếp cận dựa trên Apriori để khai phá tập phổ biến mờ, sau đó các tập mục phổ biến mờ còn lại có thể được sử dụng để tạo ra các luật kết hợp mờ. như F-APACS [69], [31], [32]. Trong đó, các giá trị của các thuộc tính định lượng đầu tiên được chuyển đổi thành biểu diễn của các thuật ngữ ngôn ngữ với các giá trị liên thuộc của chúng theo các hàm liên thuộc được xác định trước.

1.4.2 Các nghiên cứu mở rộng từ Apriori

Một số thuật toán biến thể đã được trình bày để khai phá các luật kết hợp mờ [70], [71], [72], [73], [74]. Sau đó tác giả cũng đã phát triển một thuật toán khai phá mờ nhiều mức để khai phá các luật kết hợp mờ bằng cách tích hợp các khái niệm tập mờ và phân loại nhiều mức [28].

1.4.3 Các phương pháp nghiên cứu dựa trên cây

Để giải quyết vấn đề thời gian tính toán, Papadimitriou đề xuất thuật toán cây mẫu thường xuyên mờ (FFPT- Frequent Fuzzy Pattern Tree) [75]**.** Lin sau đó trình bày một framework để khai phá mờ khác để tìm ra các mục phổ biến mờ dựa trên cấu trúc cây. Ba thuật toán là cây phổ biến mờ FP (FFP)-tree [38]**,** cây phổ biến mờ nén (CFFP)-tree [39] và cây mẫu phổ biến mờ giới hạn trên (UBFFP)-tree [40] đã được phát triển để khai phá các tập mục phổ biến mờ từ cơ sở dữ liệu định lượng. Các thuật toán này khác nhau chủ yếu ở cấu tạo cây.

1.5 Xác định vấn đề nghiên cứu

Trong các phương pháp khai phá dữ liệu mờ dựa trên Apriori, các giá trị định lượng được chuyển đổi thành các tập mờ theo các hàm thành viên được xác định trước. Sau đó, tập phổ biến mờ và luật kết hợp mờ có thể được tạo ra dựa trên quá trình thực hiện Apriori. Do thời gian thực hiện của các phương pháp dựa trên Apriori tốn nhiều thời gian nên các phương pháp khai thác dữ liệu mờ dựa trên cây được mô tả để tăng tốc quá trình khai thác. Về cơ bản, các phương pháp này được sửa đổi từ cây FP để xử lý các tập mục mờ. việc khai phá các tập mục mờ phổ biến được thực hiện hoàn toàn trên cây, điều này chiếm nhiều không gian bộ nhớ. Luận án đã đề xuất thuật toán khai phá tập mục phổ biến mờ dựa trên cấu trúc Nodelist nhằm giải quyết vấn đề về không gian bộ nhớ trong công trình [CT1], [CT2], [CT5].

Hơn nữa, các hàm thành viên có thể được đưa ra bởi các chuyên gia. Tuy nhiên, các ý kiến chuyên gia có thể không phải lúc nào cũng có sẵn. Để giải quyết vấn đề này, luận án đã đề xuất phương pháp xác định các tập mờ cho mỗi thuộc tính định lượng trong cơ sở dữ liệu bằng kỹ thuật phân cụm EMC trong công trình [CT2], [CT2], [CT4].

Cùng với sự gia tăng kích thước của dữ liệu, khai phá dữ liệu trong cơ sở dữ liệu lớn đã trở thành một vấn đề quan trọng. Bên cạnh việc áp dụng điện toán đám mây hoặc các kiến trúc song song và phân phối khác để tăng tốc quá trình khai phá mờ cũng đáng để nghiên cứu. Luận án đề xuất phương pháp xử lý song song quá trình khai phá tập mục phổ biến sử dụng hướng tiếp cận automata di động học. [CT3]

Kết luận chương 1

Trong chương 1, luận án trình bày tóm tắt tổng quan các vấn đề liên quan đến luật kết hợp, logic mờ và luật kết hợp mờ, các phương pháp khai phá dữ liệu mờ khác nhau, bao gồm khai phá dữ liệu mờ dựa trên Apriori, khai phá dữ liệu mờ dựa trên cây rồi từ đó xác định các vấn đề nghiên cứu của luận án.

Luận án này tập trung trình bày việc giải quyết các bài toán thuộc các nghiên cứu [CT1], [CT2], [CT5], CT[4], [CT3]. Cụ thể, luận án sẽ tập trung nghiên cứu đề xuất và giải pháp giải quyết triệt để 3 vấn đề sau đây:

- Đề xuất phương pháp xác định các tập mờ cho mỗi thuộc tính định lượng trong cơ sở dữ liệu thông qua kỹ thuật phân cụm.
- Đề xuất phương pháp khai phá tập mục phổ biến mờ trong cơ sở dữ liệu định lượng sử dụng cấu trúc dữ liệu Node-list.
- Đề xuất một phương pháp xử lý song song để khai phá các tập mờ phổ biến sử dụng phương pháp tiếp cận automata di động học (Cellular learning automata).
- Nội dung hai chương còn lại trong luận án sẽ trình bày giải pháp tương ứng cho ba vấn đề nghiên cứu trên.

CHƯƠNG 2: KHAI PHÁ TẬP MỤC PHỔ BIẾN MỜ DỰA TRÊN CẤU TRÚC CÂY

Ở chương này, luận án trình bày quy trình thực hiện khai phá luật kết hợp mờ. Trong đó, ở bước thứ nhất để chuyển đổi cơ sở dữ liệu định lượng sang cơ sở dữ liệu mờ, NCS thực hiện mờ hóa giá trị định lượng của các mục bằng phương pháp phân cụm EMC và xác định các khoảng mờ, kết quả của hai thuật toán này được sử dụng cho bước tiền xử lý dữ liệu sau đó áp dụng các hàm thành viên để chuyển đổi giá trị định lượng sang giá trị mờ. Bước thứ hai, NCS đề xuất hai phương pháp khai phá tập mục mờ phổ biến sử dụng cấu trúc Nodelist dựa trên cây tiền tố hậu tố (FPPC – Fuzzy Pre-order, Post-order Code) và cây FPOSC (Fuzzy Pre-order Size Code). Kết quả của bước khai phá tập mục mờ phổ biến là cơ sở chính được sử dụng để thực hiện tìm các luật kết hợp mờ.

2.1 Phát biểu bài toán khai phá luật kết hợp mờ

Cho một cơ sở dữ liệu định lượng $D_Q = \{T_1, T_2, \dots, T_n\}$ và tập các mục $I = \{I_1, I_2, \dots, I_m\}$. Cho một tập mờ $A_j = \{A_{j1}, A_{j2}, \dots, A_{jh}\}$ trong đó A_{jk} được xác định là phần tử thứ k[th] trong tập mờ A_j của mục I_j và $f_{j,k}^{(i)}$ là giá trị mờ (được xác định bởi hàm thành viên) của A_{jk} trong giao tác T_i.

Khai phá luật kết hợp mờ là vấn đề tìm ra tất cả các luật có dạng $A \to B$ thỏa mãn $fsp(A \to B) \geq minsup$ và $fcf(A \to B) \geq minconf$, với minsup và minconf lần lượt là độ hỗ trợ và độ tin cậy tối thiểu do người dùng định nghĩa. Thuật toán khai phá luật kết hợp mờ được thực hiện qua 3 pha chính.

- Pha 1: chuyển đổi cơ sở dữ liệu định lượng sang cơ sở dữ liệu mờ
- Pha 2: Khai phá các tập mục phổ biến mờ có độ hỗ trợ mờ lớn hơn ngưỡng hỗ trợ tối thiểu $FFI_k = \{A | \text{ fsup } (A) \geq \delta\}$.
- Pha 3: Khai phá tất cả luật kết hợp mờ có độ tin cậy lớn hơn ngưỡng tin cậy tối thiểu từ các mục mờ phổ biến được tìm thấy trong pha 2.

2.2 Thuật toán phân cụm dữ liệu và xác định các khoảng mờ

2.2.1 Các khái niệm cơ bản

2.2.1.1 Phân cụm dữ liệu

Trong khai phá dữ liệu phương pháp tối đa hóa kì vọng Expectation Maximization (EM) [77]–[79] là thuật toán gom nhóm dữ liệu được dùng trong tác vụ khám phá tri thức. Thuật toán EM có những điểm hạn chế như sau: Thứ nhất, EM chạy nhanh ở các vòng lặp ban đầu nhưng chậm hơn ở các vòng lặp sau. Thứ hai, EM không phải lúc nào cũng tìm được tham số tối ưu cho toàn cục, thay vào đó là tối ưu cục bộ.

Định nghĩa 2.1: Hệ số biến thiên C_v được định nghĩa là tỉ số của độ lệch chuẩn σ so với kỳ vọng $\overline{x}$ của cụm i chứa các phần tử $X_i\{x_{i_1}, x_{i_2}, \dots, x_{i_n}\}$ tức là:

$$C_v(X_i) = \frac{\sigma}{\bar{x}} \times 100 \quad (2.1)$$

Định nghĩa 2.2: Phân bố Gaussian đơn biến [77]–[79]

$$N(X|\bar{x}, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X-\bar{x})^2}{2\sigma^2}} \quad (2.2)$$

Định nghĩa 2.3: Phân bố Gaussian đa biến [77]–[79]

$$N(X|\bar{x}, \Sigma) = \frac{1}{(2\pi|\Sigma|)^{1/2}} exp\left\{-\frac{1}{2}(X-\bar{x})^T \sum^{-1}(X-\bar{x})\right\} \quad (2.3)$$

Phương pháp ước lượng hóa tham số (Maximum Likelihood). Tính log cho phân phối Gaussian. [77]–[79]

$$ln\, p(X|\bar{x}, \Sigma) = -\frac{1}{2}ln(2\pi) - \frac{1}{2}ln|\Sigma| - \frac{1}{2}(X-\bar{x})^T \sum^{-1}(X-\bar{x}) \quad (2.4)$$

Lấy đạo hàm và gắn bằng 0

$$\frac{\delta \ln p(X|\bar{x},\Sigma)}{\delta \bar{x}} = 0, \bar{x}_{ML} = \frac{1}{N}\sum_{n=1}^{N} X_n$$

$$\frac{\delta \ln p(X|\bar{x},\Sigma)}{\delta \Sigma} = 0, \qquad \Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N} X_n$$

Trong đó N số lượng các mẫu. Phân phối hỗn hợp tuyến tính của Gaussian

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X|\bar{x}_k, \Sigma_k) \quad (2.5)$$

Trong đó K số của Gaussian và π_k hệ số pha trộn, với trọng số cho mỗi đơn vị Gaussian $0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1$. Xét log likelihood

$$\ln p(X|\bar{x},\Sigma,\pi) = \sum^{N} \ln p\,(X_n) = \sum^{N} \ln\left\{\sum_{n=1}^{N} \pi_n (X_n|\bar{x}_k,\Sigma_k)\right\} \quad (2.6)$$

2.2.1.2 Xác định các khoảng mờ

Khi thao tác dữ liệu trong cơ sở dữ liệu mờ, vấn đề quan trọng nhất là làm thế nào tìm ra một phương pháp xử lý các giá trị mờ để từ đó xây dựng các quan hệ đối sánh giữa chúng. Các giá trị trong cơ sở dữ liệu mờ rất phức tạp, bao gồm các giá trị ngôn ngữ, giá trị số, giá trị khoảng. Có nhiều cách tiếp cận khác nhau để xử lý các giá trị mờ được các tác giả trong và ngoài nước quan tâm nghiên cứu trong những năm gần đây, chẳng hạn như: lý thuyết thuyết tập mờ [22], lý thuyết khả năng [80], [81], quan hệ tương tự [82]. Các giá trị khoảng hầu như chuyển về dạng các các số mờ theo các dạng như tam giác, hình thang, hình chuông để xử lý.

2.2.2 Bài toán đặt ra

Cho trước cơ sở dữ liệu chứa các giá trị định lượng D_Q.

Bài toán đặt ra: Xác định tập các tập mờ của các thuộc tính định lượng trong D_Q cùng các hàm thành viên tương ứng. Chuyển đổi cơ sở dữ liệu định lượng sang cơ sở dữ liệu mờ.

2.2.3 Thuật toán phân cụm dữ liệu EMC

2.2.3.1 Ý tưởng thuật toán

Thuật toán EMC là một kỹ thuật tối ưu hóa lặp lại được vận hành linh hoạt (Thuật toán được cải thiện để tăng tính linh hoạt cho phân cụm đồng thời giảm tối ưu hóa cục bộ và tăng tối ưu hóa toàn cục).

1) **Bước E:** dựa trên các tham số của mô hình, tính toán các xác suất gán nhãn các điểm dữ liệu vào một nhóm.
2) **Bước M**: cập nhật các tham số của mô hình dựa trên các nhóm gom được từ bước **E**.
3) **Bước C**: Cập nhật các tham số của mô hình dựa trên các biến tiềm ẩn tính theo phương pháp khả năng ước lượng cực đại và tỷ lệ tương tự giữa các đối tượng trong một cụm và đánh giá hệ số biến thiên của các phần tử trong cụm.

Thuật toán EMC bắt đầu bằng các tham số cho mô hình dự đoán. Sau đó thực hiện vòng lặp 5 tiến trình được thể hiện trong Thuật toán 2.1.

2.2.3.2 Thuật toán EMC

Thuật toán EMC được mô tả như trong Thuật toán 2.1

Thuật toán 2.1: EMC (Expectation Maximization Coefficient)

Đầu vào: Khởi tạo giá trị của hệ số biến thiên $C_{v_{value}}$

Đầu ra: Số cụm tối ưu

1: Khởi tạo các tham số kỳ vọng $\bar{x}_j$, hiệp phương sai Σ_j, hệ số pha trộn π_j trong đó $\sum_{j=1}^{j} \pi_j = 1$ và $\pi_j \geq 0 \; \forall j$. Hệ số biến thiên $C_{v_{value}} = 15\ \%$ đây là giá trị khởi tạo từ người dùng để tính toán tỉ lệ biến thiện giữa các phần tử trong một cụm và các cụm.

2: **Bước E**: Dựa trên các tham số của mô hình, tính toán các xác suất gán nhãn các điểm dữ liệu vào một nhóm

$$\gamma_j(X) = \frac{\pi_k \mathcal{N}(X|\bar{x}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(X|\bar{x}_j, \Sigma_j\right)} \tag{2.7}$$

3:

Bước M: Cập nhật các tham số của mô hình dựa trên các nhóm gom được từ bước E

$$\bar{x}_j = \frac{\sum_{n=1}^{N} \gamma_j(X_n) X_n}{\sum_{n=1}^{N} \gamma_j(X_n)} \tag{2.8}$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(X_n)\left(X_n - \bar{x}_j\right) X_n - \bar{x}_j{}^T}{\sum_{n=1}^{N} \gamma_j(X_n)} \tag{2.9}$$

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(X_n) \tag{2.10}$$

4: Đánh giá log likelihood.

$$\ln p(X|\bar{x}, \Sigma, \pi) = \sum_{n-1}^{N} \ln = \sum^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k (X_n|\bar{x}_k, \Sigma_k) \right\} \tag{2.11}$$

5:

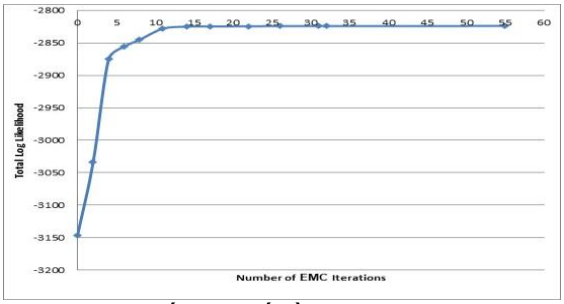
Bước C: Cập nhật thông tin về hệ số biến thiên của các cụm và đánh giá khả năng biến động các phần tử cho từng cụm, cụ thể ta đánh giá hệ số biến thiên của cụm thứ i với C_{v_i} có thảo mãn giá trị biến thiên $C_{v_{value}}$ đã cho hay không.

$$C_{v_i} = \frac{\sum_{n=1}^{N} \gamma_j(X_n) X_n}{\frac{1}{n} \sum_{k=1}^{n} x_k} \tag{2.12}$$

$$C_{v_i} \leq C_{v_{value}} \tag{2.13}$$

6: Nếu không hội tụ và thảo mãn giá trị biến thiên $C_{v_{value}}$ đã cho, quay trở lại bước 2. Nếu likelihood không có nhiều thay đổi thuật toán kết thúc.

2.2.3.3 Đánh giá thuật toán EMC dựa trên Log Likehood



Hình 2.1: Tính tổng Log Likelihood đối với số lần lặp lại của thuật toán EMC

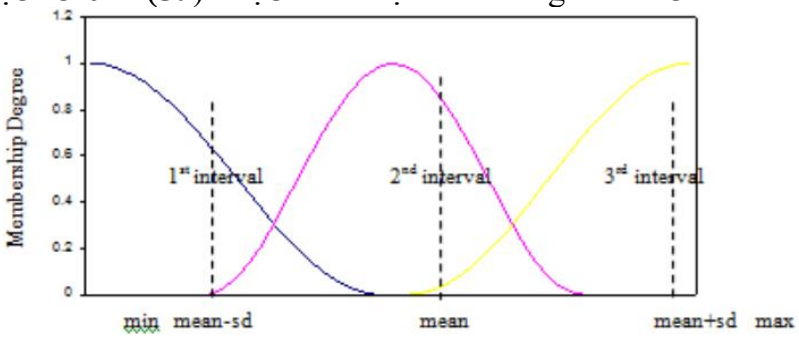
Thông qua kết quả thực nghiệm trong Hình 2.2, trong vùng giá trị (Total Log Likelihood (TLL)> -3150) của TLL, ta có thể tìm thấy kết quả tốt nhất từ tham số cho mô hình GMM. Các giá trị tính toán của C_v khác nhau tương ứng với từng cụm ảnh hưởng đến số lần lặp EMC rất nhiều. Giá trị của một C_v có thể thay đổi linh hoạt, điều này phụ thuộc vào số lượng cụm cũng như kích thước của mỗi cụm. Kết quả thu được

từ thuật toán sẽ cho ra các cụm tối ưu và sử dụng chúng để phân loại từng thuộc tính định lượng thành một tập mờ bằng việc xác định các hàm thành viên.

2.2.4 Thuật toán xác định các khoảng mờ

2.2.4.1 Xác định tâm

Trong cơ sở dữ liệu mờ, miền giá trị của các thuộc tính định lượng của mục mờ mà trong đó (các thuộc tính có thể chứa giá trị rõ hoặc mờ) được chia thành hai hoặc nhiều khoảng mờ. Trong các khoảng mờ, một phần tử có thể thuộc nhiều hơn một khoảng với các mức độ khác nhau. Trong phần mục này, giả sử mỗi thuộc tính định lượng được chia thành ba khoảng mờ bằng phương pháp tiếp cận thống kê trong đó sử dụng kỳ vọng $\bar{x}$ (mean) và độ lệch chuẩn (*Sd*) được minh họa như trong hình 2.3.



Hình 2.2: Các khoảng mờ

Mức độ chồng lấp giữa các đối tượng dữ liệu mờ thuộc về hai hoặc nhiều cụm được định nghĩa như sau:

$$Overlap = \frac{\sum_{j=1}^{n}|C_j|}{|\cup_j^n C_j|} * 100 \qquad (2.14)$$

Trong đó C_j là cụm thứ *j*, với *j=1, 2,..., n;*

2.2.4.2 Xác định các khoảng mờ

Khoảng thứ nhất (1^{st} interval) Biên dưới (d^-) của khoảng thứ nhất là giá trị nhỏ nhất trong miền của thuộc tính định lượng. Biên trên (d^+) được tính bằng kỳ vọng $\bar{x}$ và độ lệch chuẩn (Sd) của các giá trị của thuộc tính định lượng. Biểu thức toán học của (d^-) và (d^+) được trình bày như sau:

$$\left.\begin{aligned} d^- &= MIN(X_{1Cj}, X_{2Cj}, \dots, X_{NCj}) \\ d^+ &= \bar{x} - \frac{Sd}{2} + \bar{x} \times overlap \end{aligned}\right\} \qquad (2.15)$$

Trong đó X_N là giá trị của cụm C_j với $N = 1,2,..,n$ và $j = 1,2,..n$.

Trong khoảng thứ nhất (1^{st} interval) hàm thành viên Z-membership được sử dụng để tính mức thành viên, đó là:

$$f(x)_Z = \frac{1}{2} + \frac{1}{2}\, cos\left(\frac{x - d^-}{d^+ - d^-}\right)\Pi \qquad (2.16)$$

Khoảng thứ hai (2^{st} interval) Biên dưới (d^-) và biên trên (d^+) của khoảng thứ hai được tính như sau:

$$\left.\begin{aligned} d^- &= \bar{x} - \frac{Sd}{2} - \bar{x} * overlap \\ d^+ &= \bar{x} + \frac{Sd}{2} + \bar{x} * overlap \end{aligned}\right\} \qquad (2.17)$$

Khoảng này sử dụng cả hàm thành viên S-membership và Z-membership, được biểu diễn như sau:

$$\left.\begin{aligned} f(x)_S &= \frac{1}{2} + \frac{1}{2}\, cos\left(\frac{\bar{x} - x}{\bar{x} - d^-}\right)\Pi, \text{với } d^- \le x \le \bar{x} \\ f(x)_Z &= \frac{1}{2} + \frac{1}{2}\, cos\left(\frac{x - \bar{x}}{d^+ - \bar{x}}\right)\Pi, \text{với } \bar{x} \le x \le d^+ \end{aligned}\right\} \qquad (2.18)$$

Khoảng thứ ba (3^{st} interval) Biên dưới (d^-) và biên trên (d^+) của khoảng thứ ba được tính như sau:

$$\left.\begin{aligned} d^- &= \bar{x} - \frac{Sd}{2} - \bar{x} * overlap \\ d^+ &= MAX(X_{1Cj}, X_{2Cj}, \dots., X_{NCj}) \end{aligned}\right\} \quad (2.19)$$

Khoảng này sử dụng hàm thành viên S-Membership có dạng như sau.

$$f(x)_S = \frac{1}{2} + \frac{1}{2}\, cos\left(\frac{d^+ - x}{d^+ - d^-}\right)\Pi \quad (2.20)$$

2.2.4.3 Chuyển đổi CSDL định lượng sang CSDL mờ

Sau khi xác định các khoảng mờ, cơ sở dữ liệu định lượng ban đầu được chuyển đổi thành cơ sở dữ liệu mờ, chuẩn bị cho quá trình khai phá luật kết hợp mờ. Đối với mỗi tập mờ mà chúng ta đã xác định trước đó, có một hàng trong cơ sở dữ liệu mới chứa mức độ thành viên của các phần tử đơn lẻ đối với tập cụ thể.

2.3 Khai phá tập mục phổ biến mờ

2.3.1 Bài toán đặt ra

Cho cơ sở dữ liệu chứa các giá trị mờ D_f và độ hỗ trợ tối thiểu δ

Bài toán đặt ra: Tìm các tập mục phổ biến mờ có dạng:

$$FFI_k := \{X |\, sup(X) \geq \delta \times |D_f|\}$$

2.3.2 Khai phá tập mục phổ biến mờ sử dụng cấu trúc cây FPPC-tree

2.3.2.1 Ý tưởng thuật toán

Từ CSDL chứa giá trị mờ D_f, tính độ hỗ trợ của mỗi mục mờ A_{il} trong giao tác T_q. Kiểm tra nếu độ hỗ trợ của mục mờ A_{il} lớn hơn độ hỗ trợ tối thiểu δ thì thêm A_{il} vào F_1. Sắp xếp các mục phổ biến mờ trong F_1 theo độ hỗ trợ giảm dần. Các mục mờ không phải là mục phổ biến mờ thì loại khỏi D_f. Xây dựng cây FPPC.

Sau khi xây dựng cây FPPC, bằng cách duyệt qua cây FPPC theo thứ tự pre-order, ta thu được Nodelist của mỗi mục phổ biến mờ (1-item). Với mỗi nút N_i, ta chèn $\langle N_i.pre, N_i.post, N_i.support \rangle$ vào Nodelist của mỗi mục được đại diện bởi N. Cây FPPC được xóa đi sau khi thu được Nodelist nhằm giảm không gian bộ nhớ.

Sau khi có được Nodelist của mỗi mục phổ biến 1-item, ta thực hiện giao Nodelist của các mục phổ biến 1-item để tìm Nodelist của tập mục (k-itemset). với bất kỳ ứng cử viên (k + 1) P_c nào, ta có được độ hỗ trợ của P_c bằng cách tính tổng giá trị độ hỗ trợ của tất cả các FPP_Code trong Nodelist của nó. Dựa vào độ hỗ trợ của P_c, chúng ta có thể đánh giá liệu P_c có phổ biến hay không. Bằng cách lặp lại quy trình trên, ta tìm được tất cả các mẫu mờ phổ biến.

2.3.2.2 Thuật toán xây dựng cây FPPC

Thuật toán xây dựng cây FPPC được mô tả như trong Thuật toán 2.2

Thuật toán 2.2: Xây dựng cây FPPC_tree

Input: CSDL chứa giá trị mờ D_f, độ hỗ trợ mờ tối thiểu fminsup δ.

Output: FPPC-tree (FTr), tập mục mờ phổ biến 1-itemset (F_1).

(1) Duyệt qua CSDL mới D_f chứa giá trị mờ để tính độ hỗ trợ của mỗi mục mờ A_{il} trong giao tác T_q theo công thức:

$$sup(A_{il}) = \sum_{A_{il} \subseteq T_q \wedge T_q \in D_f} f_{il}$$

(2) Nếu $sup(A_{il}) \geq minsup\ \delta$, thêm A_{il} vào F_1. Ta có $F_1 = \{A_{il} \mid sup(A_{il}) \geq n \times \delta\}$.

(3) Sắp xếp các mục mờ phổ biến trong F_1 theo độ hỗ trợ giảm dần.

(4) Nếu A_{il} $not\ in$ F_1, xóa A_{il} khỏi tất cả các T_q $(q = 1..n)$.

(5) Tạo nút root của cây FPPC và đánh nhãn "null"

(6) for each T_q in D_f {

(7) Sắp xếp các mục phổ biến còn lại theo độ hỗ trợ giảm dần.
(8) Chèn các mục mờ vào cây FFPC_tree (quy trình tương tự với MFFP_tree [14])
(9) }
(10) Duyệt cây FPPC để sinh PP_Code cho mỗi nút trên cây.

2.3.2.3 Thuật toán xây dựng Nodelist của các mục phổ biến mờ dựa trên cây FFPC

Thuật toán xây dựng Nodelist của mỗi mục phổ biến mờ (1-item) được mô tả trong thuật toán 2.3

Thuật toán 2.3: Nodelist_Construction

Input: FPPC-tree (R) and $L1$ (Tập các mục mờ phổ biến 1-item)
Output: NL_1 (Tập các Node list của L_1)
1: Tạo NL_1, $NL_1[k]$ là Nodelist của $L_1[k]$
2: **for each** node N_i in R duyệt theo tiền thứ tự **do**
3: **if** $N_i.f\_item = L_1[k].f\_item$ **then**
4: insert <*N.pre, N.post, N. support*> into NL_1[k]
5: **end if**
6: **end for**

- ***Giao của Nodelist***

Thuật toán thực hiện giao Nodelist của 2 tập phổ biến mờ có độ dài k được mô tả trong thuật toán 2.4.

Thuật toán 2.4: Thuật toán FNodelist_Intersection

Input: NL_1 và NL_2 trong đó NL_1 , NL_2 là các Nodelist của 2 tập mờ phổ biến có độ dài k.
Output: NL_3 là Nodelist của tập mờ phổ biến có độ dài (k+1) .

(1) for $(i = 0; i < NL_1.Size(); i++)$ do
(2) for $(j = 0; i < NL_2.Size(); j++)$ do
(3) if $(NL_1[i].fpre\_code < NL_2[j].fpre\_code)$ then
(4) if $(NL_1[i].fpos\_code > NL_2[j].fpos\_code)$ then
(5) Thêm $NL_2[j]$ vào NL_3;
(6) End if
(7) else
(8) if $(NL_1[i].fpos\_code < NL_2[j].fpos\_code)$ then
(9) Thêm $NL_1[i]$ vào NL_3;
(10) End if
(11) End if
(12) End for
(13) return NL_3
(14) End for

2.3.2.4 Thuật toán NFFP

Thuật toán NFFP được mô tả như trong Thuật toán 2.5

Thuật toán 2.5: Thuật toán khai phá tập mục mờ phổ biến NFFP

Input: độ hỗ trợ mờ tối thiểu *fminsup* (δ), tập mờ phổ biến (1-item) (L_1), Nodelist của $L1$ ($NL1$);

Output: Tập mục mờ phổ biến (FFIs)

(1) For $(k = 2; L_{k-1} \neq \emptyset; k++)$ do begin
(2) For each $p = i_1 i_2 \dots i_{k-2} i_x \in L_{k-1}$ and $q = i_1 i_2 \dots i_{k-2} i_y \in L_{k-1}$, do
(3) If $i_x \succ i_y$ then
(4) $l = i_1 i_2 \dots i_{k-2} i_x i_y$
(5) If each *k*-1 subsets l in L_{k-1} then begin
(6) *l*.Node-list = NL_Intersection (*p*.Node-list, *q*.Node-list);
(7) Tính $l.support$; // Sử dụng tính chất 2.4
(8) If $(l.support \geq n \times \delta)$ then begin
(9) $L_k = L_k \cup \{l\}$;
(10) $NL_k = NL_k \cup \{l.Nodelist\}$;
(11) end if
(12) end if
(13) end if
(14) end for
(15) Xóa NL_{k-1};
(16) end for
(17) $FFIs = \bigcup_k L_k$

2.3.3 Khai phá tập mục phổ biến sử dụng cấu trúc cây FPOSC-tree

2.3.3.1 Ý tưởng thuật toán

Từ CSDL chứa giá trị mờ D_f, tính độ hỗ trợ của mỗi mục mờ A_{il} trong giao tác T_q. Kiểm tra nếu độ hỗ trợ của mục mờ A_{il} lớn hơn độ hỗ trợ tối thiểu δ thì thêm A_{il} vào F_1. Sắp xếp các mục phổ biến mờ trong F_1 theo độ hỗ trợ giảm dần. Các mục mờ không phải là mục phổ biến mờ thì loại khỏi D_f. Xây dựng cây FPOSC.

Trong khi xây dựng cây FPOSC có thể thêm một số nút con mà không cần phải duyệt lại cây và pre-order được tính toán cùng lúc với việc xây dựng Node-list của các mục mờ phổ biến. Với mỗi nút N_i, ta chèn $\langle N_i.pre, N_i.size, N_i.f\_sup \rangle$ vào Nodelist của mỗi mục được đại diện bởi N. Cây FPOSC được xóa đi sau khi thu được Nodelist nhằm giảm không gian bộ nhớ.

Sau khi có được Nodelist của mỗi mục phổ biến 1-item, ta thực hiện giao Nodelist của các mục phổ biến 1-item để tìm Nodelist của tập mục (k-itemset). với bất kỳ ứng cử viên (k + 1) P_c nào, ta có được độ hỗ trợ của P_c bằng cách tính tổng giá trị độ hỗ trợ của tất cả các FPP_Code trong Nodelist của nó. Dựa vào độ hỗ trợ của P_c, chúng ta có thể đánh giá liệu P_c có phổ biến hay không. Bằng cách lặp lại quy trình trên, ta tìm được tất cả các mẫu mờ phổ biến.

2.3.3.2 Thuật toán xây dựng cây FPOSC (Fuzzy Pre-order Size Coding)

Thuật toán xây dựng cây FPOSC được xác định bằng cách điều chỉnh cấu trúc của cây FPPC [CT1], được trình bày trong thuật toán 2.6.

Thuật toán 2.6: FPOSC-Tree_Construction
Input: Fuzzy Database D_f, fminsup δ
Output: FT_r (FPOSC-tree), F_1 (frequent fuzzy itemsets (length=1)
Begin
1: Duyệt D_f tính độ hỗ trợ của mỗi mục mờ A_{jk} trong giao dịch T_i
2: If $fsup(A_{jk}) \geq \delta$ then
3: Thêm A_{jk} vào F_1;
4: End if
5: If A_{jk} not in F_1 then
6: Xóa A_{jk} ra khỏi tất cả T_i $(i = 1 \dots n)$
7: End if
8: Tạo FT_r NodeRoot=null
9: Đặt Flist là danh sách chứa các mục mờ còn lại trong mỗi T_i

10: For each T_i in D_f
11: Sắp xếp FList theo thứ tự fsup giảm dần
12: $e = FList[0]$; e là phần tử đầu tiên trong Flist
13: $List_r = List[size - 1]$
14 Insert_tree ($[e| List_r], FT_r$)
15: End for

/* Thủ tục Insert_Tree được sử dụng để gọi đệ quy việc xây dựng cây POSC. Trong đó, e là phần tử đầu tiên của Flist và Flist là danh sách còn lại */

Procedure Insert_tree ($[e| List_r], FT_r$)
1: Gọi N là một nút tương ứng với một nhánh trong FT_r
2 If $e.fitem == N.fitem$ then
3: Cộng giá trị mờ $f_{j,k}^{(i)}$ của e vào fsup của N
4: Else
5: Tạo nút mới N có fsup là $f_{j,k}^{(i)}$ và thêm N vào cuối nhánh tương ứng.
6: $N.size = 1$
7: If $List_r$ is nonempty then
8: Gọi Insert_Tree ($List_r$,N) recursively
9: End if
10: End if
11: $N.size = N.countChild + 1$
End procedure

2.3.3.3 Thuật toán xây dựng Nodelist của các mục phổ biến mờ dựa trên cây FPOSC

Thuật toán xây dựng Node-list của tập mục phổ biến mờ (length=1) F_1 được trình bày trong Thuật toán 2.7.

Thuật toán 2.7: FNode_List_Gen
Input: POSC-tree (FT_r), tập mục phổ biến mờ length=1 (F_1)
Output: Node-list của F_1 (NL_1)
Begin
1: for each N_i in FT_r (được duyệt theo thứ tự trước trong FT_r),
2: Gọi $NL_1[k]$ là Node-list của mục k^{th} trong F_1.
3: If $N_i.f_{item} == F_1[k].f\_item$ then
4: Thêm $\langle N_i.pre, N_i.size, N_i.fsup\rangle$ vào $NL_1[k]$;
5: Return $NL_1 = \bigcup_k NL_1[k]$
End.

Phương thức xây dựng giao của hai Node-list được thực hiện trong thuật toán POS Node-list Intersect.

Thuật toán 2.8: POS_Node-list_Intersect
Input: NL_{k1}, NL_{k2} trong đó NL_{k1}, NL_{k2} là Node-list của 2 tập mục mờ k-itemsets
Output: NL_{k1+1} – Node-list của tập mục mờ (k+1) itemsets
Begin
1: for $i = 0; i < NL_{k1}.length; i++$ do
2: For $j = 0; j < NL_{k2}.length; j++$ do
3: If $NL_{k1}[i].pre < NL_{k2}[j].pre$ then
4: If $NL_{k2}[j].pre < NL_{k1}[i].pre + NL_{k1}[i].size$ then
5: Thêm $NL_{k2}[j]$ vào NL_{k1+1}
6: End if
7: else
8: If $NL_{k1}[i].pre < NL_{k2}[j].pre + NL_{k2}[j].size$ then
9: Thêm $NL_{k1}[i]$ vào NL_{k1+1}

10: End if
11: End if
12: End for
13: End for
End.

2.3.3.4 Thuật toán NPSFF

Thuật toán *2*.9: Thuật toán khai phá tập mục mờ phổ biến NPSFF

Input: độ hỗ trợ mờ tối thiểu *fminsup* (δ), tập mờ phổ biến (1-item) (L_1), Nodelist của $L1$ ($NL1$);
Output: Tập mục mờ phổ biến (FFIs)

1: For $(k = 2; L_{k-1} \neq \emptyset; k++)$ do begin
2: For each $p = i_1 i_2 \dots i_{k-2} i_x \in L_{k-1}$ *and* $q = i_1 i_2 \dots i_{k-2} i_y \in L_{k-1}$, do
3: If $i_x \succ i_y$ then
4: $l = i_1 i_2 \dots i_{k-2} i_x i_y$
5: If each *k*-1 subsets l in L_{k-1} then begin
6: *l*.Node-list = POS_Node-list_Intersect (*p*.Node-list, *q*.Node-list);
7: Tính $l.support$; // Sử dụng tính chất 2.4
8: If $(l.support \geq n \times \delta)$ then begin
9: $L_k = L_k \cup \{l\}$;
10: $NL_k = NL_k \cup \{l.Nodelist\}$;
11: End if
12: End if
13: End if
14: End for
15: Delete NL_{k-1};
16: End for
17: $FFIs = \bigcup_k L_k$;

2.4 Thuật toán khai phá luật kết hợp mờ

Thuật toán 2.10: MFAR

Input: CSDL định lượng (D_Q), ngưỡng độ hỗ trợ tối thiểu δ, độ tin cậy tối thiểu minfc

Output: Tất cả các luật kết hợp mờ FRs

Begin

1: Chuyển đổi D_Q sang D_f
2: Thực hiện FPOSC _Tree_ Construction (*Df*, δ) to sinh FPOSC Tree (FTr), F_1
3: Thực hiện FNode-list Gen (FTr, F_1)
4: Thực hiện NPSFF (δ, L_1, NL_1) để tìm ra tất cả mục mờ phổ biến
5: $FRs = \emptyset$;
6: For each $X \in FFIs$ do
7: For each $Y \subset X$ && $Y \neq \phi$ do
8: $fr = X \setminus Y \rightarrow Y$;
9: $fc(fr) = \frac{sup(XY)}{sup(X)}$;
10: If $fc(fr) \geq mincf$ then
11: $FRs = FRs \cup \{fr\}$;
12: End if
13: End for
14: End for
15: Return FRs

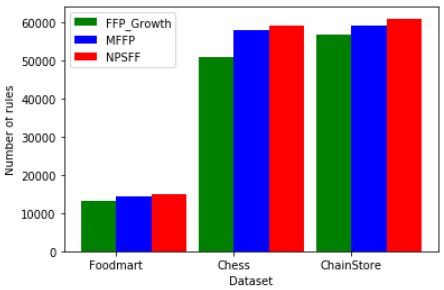
2.5 Thực nghiệm

Trong thử nghiệm, NCS sử dụng tập dữ liệu thu được từ tập dữ liệu để khai phá tập mục phổ biến được gọi là Foodmart, Chess và Chain [79]. Mỗi giao dịch trong tập dữ

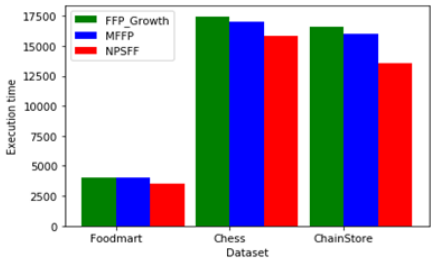
liệu này bao gồm tất cả các mặt hàng mà khách hàng có được trong một lần. Mô tả về tập dữ liệu được trình bày trong bảng 2.10. Để xử lý cơ sở dữ liệu định lượng, NCS đã chỉ định số ngẫu nhiên cho tất cả các mục trong tập dữ liệu này với phân phối trong phạm vi giá trị từ 1 đến 100.

Bảng 2.10: Mô tả tập dữ liệu cho thực nghiệm

Tập dữ liệu	**Số giao dịch**	**Số mục**	**Số lượng mục trung bình trên mỗi giao dịch**
Foodmart	4,141	1,559	4.42
Chess	3,196	75	37
ChainStore	111,294	46,086	7.23



Hình 2.3: Số luật sinh ra từ 3 thuật toán



Hình 2.4: Thời gian thực thi của các thuật toán

2.6 Kết luận chương 2

Trong chương này, NCS đưa ra giải pháp giải quyết các vấn đề liên quan đến ranh giới "sắc nét" giữa các khoảng mờ bằng cách đề xuất 2 thuật toán phân cụm dữ liệu EMC và phân cụm ái lực FAP. Kết quả của 2 thuật toán này được sử dụng trong giai đoạn tiền xử lý dữ liệu, phân vùng dữ liêu để chuyển đổi cơ sở dữ liệu định lượng sang cơ sở dữ liệu mờ. Thứ hai, luận án đưa ra 2 phương pháp khai phá luật kết hợp mờ dựa trên cấu trúc dữ liệu Nodelist là NFFP và NPSFF.NCS đề xuất hai thuật toán NFFP, NPSFF sử dụng cây FPPC_tree, cây POSC-tree để lưu trữ cơ sở dữ liệu định lượng với các giá trị thành viên theo thứ tự giảm dần. Dựa trên cây được xây dựng, Nodelist của từng mục mờ phổ biến được tạo ra. Sau đó, thuật toán NFFP, NPSFF thu được Nodelist của các mục mờ phổ biến (k-itemset) bằng cách giao với Nodelist của các mục k mờ phổ biến và sau đó trích xuất các tập tin mờ (k-itemset) phổ biến. Ưu điểm của thuật toán này là cây FPPC_Tree cũng như cây POSC-tree được sử dụng để tạo mã FPP_Code hoặc POS-code cho mỗi nút để lấy Nodelist của từng mục mờ phổ biến và sau đó nó sẽ bị xóa để có thể giảm yêu cầu sử dụng bộ nhớ. Do đặc tính của các mô hình dữ liệu này là sản xuất không giới hạn và tốc độ cao, những dữ liệu này không thể được lưu trữ trong bộ nhớ và vì lý do này, cần phải phát triển các kỹ thuật cho phép chúng được xử lý song song quá trình khai phá tập mục mờ phổ biến.

CHƯƠNG 3: KHAI PHÁ TẬP MỤC PHỔ BIẾN MỜ SỬ DỤNG PHƯƠNG PHÁP XỬ LÝ SONG SONG

Trong chương này, NCS trình bày một phương pháp xử lý song song để khai phá các tập mục mờ phổ biến, một giai đoạn quan trọng trong khai phá luật kết hợp mờ bằng cách sử dụng phương pháp tiếp cận tự động học di động (Cellular Learning Automata). Theo CLA, không gian được biểu diễn như một mạng, với mỗi phần tử là một ô, từng dòng một, dữ liệu giao dịch sẽ được đọc và đồng thời được chuyển đến các ô, chúng sẽ cộng tác với nhau song song. Với việc không sử dụng quy tắc vùng lân cận, một loại tự động dữ liệu được gọi là tự động học di động bất thường (ICLA) được sử dụng để tạo danh sách vùng lân cận cho mỗi ô. Thông qua việc sử dụng các automata dữ liệu di động này, việc khai thác tập mờ phổ biến được thực hiện. Quá trình này rút ngắn thời gian thực thi của thuật toán.

3.1 Giới thiệu

Trong những năm gần đây, nhiều thuật toán đã được phát triển để nghiên cứu các vấn đề về khai phá song song cho các luật kết hợp, phân loại, phân cụm và các tác vụ khác. Agrawal và cộng sự. đã đề xuất thuật toán khai phá song song đầu tiên luật kết hợp [85]–[88], trong khi Wang đã nghiên cứu các thuật toán khai phá luật kết hợp song song khác [89]–[91]. Trong số các kiến trúc song song, kiến trúc chủ-tớ (master-slave) thường được sử dụng. Hướng tiếp cận này mang lại những lợi ích đáng kể về hiệu suất [92]. Bộ xử lý chính phân bổ các nhiệm vụ cho các bộ xử lý phụ và thu thập kết quả từ chúng. Một số nghiên cứu sử dụng kiến trúc song song slave-master để thực hiện khai phá luật kết hợp phù hợp với các tập dữ liệu dày đặc như [94] [95] .

Trong lĩnh vực trích xuất luật kết hợp và PSO, các nhà nghiên cứu đã đề xuất nhiều thuật toán tính toán song song [96], [97], [98]. Đối với một lượng lớn dữ liệu thử nghiệm, một thuật toán PSO song song được áp dụng để trích xuất luật kết hợp là một giải pháp khả thi.

Ngoài ra, thuật toán iMFFP [99] đề xuất tích hợp các cây MFFP [41] khác nhau từ các cơ sở dữ liệu nhánh và được tích hợp vào cây iMFFP theo trình tự. Sau đó, Header_table được tạo ra và quá trình khai phá tập mục phổ biến được thực hiện. Với phương pháp này, việc tính toán độ hỗ trợ mờ của các mục mờ sẽ không chính xác, không đầy đủ thông tin do cơ sở dữ liệu bị phân rã. Hơn nữa, việc xây dựng từng nhánh cây MFFP và tích hợp dần vào cây iMFFP hoàn chỉnh sẽ gây tốn không gian bộ nhớ.

Trong chương này, NCS trình bày một phương pháp xử lý song song để khai phá các tập mục mờ phổ biến, một giai đoạn quan trọng trong khai phá luật kết hợp mờ bằng cách sử dụng phương pháp tiếp cận tự động học di động (Cellular Learning Automata). Trong chiến lược này, cơ sở dữ liệu định lượng ban đầu được chuyển thành cơ sở dữ liệu mờ trong bước tiền xử lý. Sau khi trích xuất tập phổ biến mờ 1-item từ tập dữ liệu, các mục mờ không phổ biến sẽ bị loại bỏ. Môi trường CA sẽ bắt đầu hoạt động sau giai đoạn tiền xử lý và tạo các ô CA khớp với từng mục mờ phổ biến 1-item. Mỗi dòng dữ liệu trong cơ sở dữ liệu nén được đọc và gửi đến các ô đồng thời, sau đó chúng hoạt động song song.

3.2 Một số khái niệm liên quan về automata di động học (Cellular learning automata)

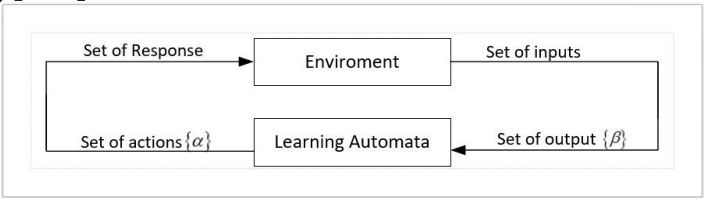
3.2.1 Automata học LA (Learning Automata)

Một LA bao gồm hai phần:

1. Một automata ngẫu nhiên với số lượng hành động hạn chế và một môi trường ngẫu nhiên.

2. Thuật toán học: thuật toán mà automata sẽ học hành động tối ưu bằng cách sử dụng hành động đó.

Mỗi hành động được chọn bởi môi trường tiềm năng sẽ được đánh giá và câu trả lời được đưa ra cho một dữ liệu tự động học. LA sẽ sử dụng câu trả lời này và chọn hành động của nó cho giai đoạn tiếp theo. Hình 3.1 cho thấy mối quan hệ giữa dữ liệu tự động học và môi trường [101].



Hình 3.1: Môi trường, LA và mối quan hệ giữa chúng

3.2.2 Automata di động (CA – Cellular Automata)

Một automata di động d-chiều [35] là một cấu trúc $A = (Z^d, \Phi, N, F)$ trong đó:

- Z^d là một mạng lưới d-tuples các số nguyên mà các mạng này có thể là vô hạn, hữu hạn hoặc bán hữu hạn.
- $\Phi = \{1, \dots, m\}$ là một tập hữu hạn các đỉnh
- $N = \{x_1, x_2, \dots, x_m\}$ là tập con hữu hạn Z^d gọi là vector láng giềng $(x_i \epsilon\, Z^d)$.
- F là quy tắc cục bộ của automata di động. Quy tắc này có thể được xác định bởi người dùng. Quy tắc các automata di động xác định các trạng thái thay đổi và cách tập hợp các ô làm láng giềng của nhau.

3.2.3 Automata di động học – Cellular learning automata

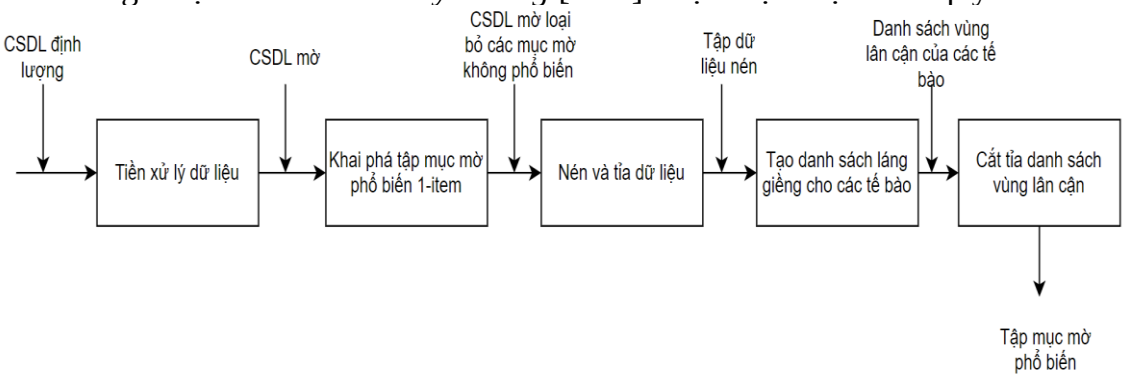
Automata di động học là sự kết hợp của hai mô hình LA và CA. Một CLA đa chiều có cấu trúc: $A = (Z^d, \Phi, A, N, F)$

- Z^d là một mạng lưới d-tuples các số nguyên mà các mạng này có thể là vô hạn, hữu hạn hoặc bán hữu hạn.
- $\Phi = \{1, \dots, m\}$ là một tập hữu hạn các đỉnh
- A là tập hợp các tự động học (LA), mỗi ô trong số đó được gán cho một ô của CLA. Mỗi ô có thể có LA hoặc nhiều hơn một.
- $N = \{x_1, x_2, \dots, x_m\}$ là tập con hữu hạn Z^d gọi là vector láng giềng $(x_i \epsilon\, Z^d)$. F là quy tắc cục bộ của automata di động.

3.3 Thuật toán khai phá tập mục phổ biến mờ sử dụng CLA

3.3.1 Ý tưởng thuật toán

Trong thuật toán *CLA-FuzzyMining* [CT3] được thực hiện theo quy trình dưới đây:



Hình 3.2: Quy trình thực hiện thuật toán CLA-Fuzzy Mining

3.3.2 Tiền xử lý dữ liệu

Trong bước này, CSDL được chuyển đổi từ CSDL định lượng sang CSDL mờ.

3.3.3 Khai phá tập mục phổ biến mờ 1-item

Việc khai phá tập mục phổ biến mờ 1-item được thực hiện như các thuật toán ở chương 2. Độ hỗ trợ mờ của mỗi mục trong giao dịch được tính theo công thức và được kiểm tra với độ hỗ trợ tối thiểu.

3.3.4 Khai phá tập mục phổ biến n-itemset

- **Thực hiện nén dữ liệu**

Thuật toán nén dữ liệu được thể hiện trong Thuật toán 3.1

Thuật toán 3.1: Data_Compression()

Input: $minsup$: độ hỗ trợ tối thiểu

Output: CDS: CSDL nén

Begin

1: **for** $i = 1$ to D_f **do**

2: For $j = 1$ *to items* **do**

3: **If** $items(i, j) == items(i + 1, j)$ **then**

4: Remove (rows (i+1));

5: Update support (rows(i)+ rows(i+1));

6: End if

7: End for

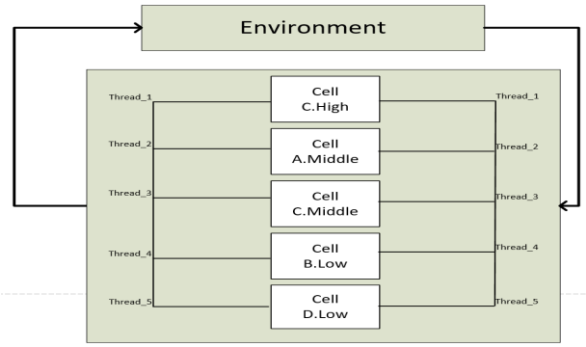
8: End for

9: Return CDS

End.

- **Xác định danh sách láng giềng**

Các ô của automata di động được tạo từ tập mục mờ phổ biến 1-item. Vì automata di động học bất quy tắc (ICLA) nên không có quy tắc cụ thể trong vùng lân cận của ô, cấu trúc vùng lân cận thông thường như Von Neumann hoặc Moore không thể được áp dụng. Môi trường CA đọc lần lượt tất cả các hàng của tập dữ liệu và sau đó chuyển chúng đến các ô trong mỗi bước. Sau khi nhận được một hàng tập dữ liệu nén, các ô sẽ bắt đầu hoạt động của chúng đồng thời với các ô khác. Các ô này sẽ cập nhật danh sách vùng lân cận của chúng tùy thuộc vào các mục mờ trong giao dịch đã nhận.



Hình 3.3: Các automata di động học theo tập mục mờ phổ biến 1-item

- **Cắt tỉa danh sách vùng lân cận**

Khi tất cả các giao dịch được môi trường gửi đến các ô, mỗi ô sẽ xóa các láng giềng và vùng lân cận có độ hỗ trợ nhỏ hơn ngưỡng tối thiểu đã được người dùng xác định khỏi danh sách vùng lân cận của nó. Danh sách vùng lân cận lại được sử dụng để quét và cuối cùng thu được tập mờ phổ biến k-Itemset. Nếu các mục này đã có trong danh sách này, chúng sẽ bị loại bỏ; nếu không, các mục này sẽ được đưa vào trong danh sách các mục mờ phổ biến.

3.3.5 Thuật toán CLA-FuzzyMining

Thuật toán CLA-FuzzyMining được mô tả như trong Thuật toán 3.2

Thuật toán 3.2: CLA_Fuzzy_Mining

Input: $minsup$: độ hỗ trợ tối thiểu

F_1: tập mục phổ biến 1-item

D_f: CSDL mờ sau khi loại bỏ các tập mục không phổ biến

CDS: CSDL nén

Output: $FFIL$: Danh sách các tập mục mờ phổ biến

Begin

1: **for** $i = 1$ to CDS **do**

3: ***CLA_Thread();***

4: End for

```
5:      Khởi tạo FFIL;
6:      for i=1 to automata cells do
7:          Thực hiện PruneNeighbors() for cell[i];
8:          Thực hiện DFS() function for cells[i];
9:          for each anItemset on cell[i].FrequentItemset do
10:             if anItemset does not exist in FFIL then
11:                 FFIL.add (anItemset);
12:             else
13:                 Nothing;
14:             End if
15:         End for
16:     End for
17:     Return FFIL;
End.
```

Hàm CLA_Thread() được mô tả trong Thuật toán 3.3.

Thuật toán 3.3: CLA_Thread()

Input: Recodset (bản ghi dữ liệu nén), NodeParent[Cell] (đại diện của các cell)

Output: automata cells

Begin

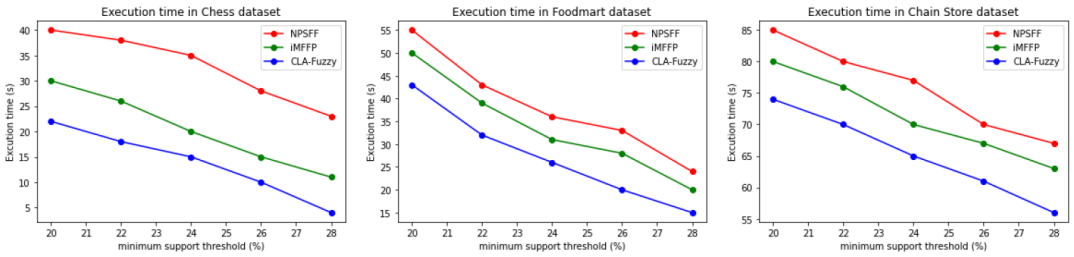
```
1:      Thread theard=new Thread();
2:      thread.Start();
3:      Initialize nodeChil=new Node();
4:      for i = 1  to Recodset do
5:          nodeChil.data= Recodset[value];
6:          If(nodeChil in (Recodset)) then
7:                  nodeChil.data= Recodset[value]+ nodeChil.data;
8:          else
9:                  NodeParent[Cell].next= nodeChil;
10:         End if
11:     End for
12:     Return AutomataCells;
End.
```

3.4 Thực nghiệm

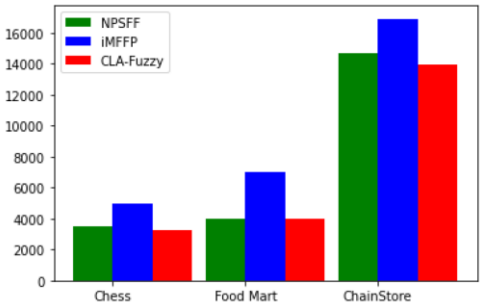
Trong phần thực nghiệm NCS sử dụng bộ dữ liệu cửa hàng Foodmart, Chess và ChainStore từ bộ dữ liệu khai phá tập phổ biến [69] cho thử nghiệm này. Mô tả của tập dữ liệu được hiển thị trong bảng 3.7. Thực nghiệm này giới thiệu các kết quả thử nghiệm từ các thuật toán và so sánh chúng với kết quả của thuật toán NPSFF [CT2] và thuật toán iMFFP [33]. Thuật toán CLA- Fuzzy Mining có hiệu quả hơn hai thuật toán trước về thời gian xử lý và bộ nhớ lưu trữ tạm thời, theo kết quả thử nghiệm dựa trên tập dữ liệu được trình bày trong bảng 7.

Bảng 3.7: Bảng dữ liệu thực nghiệm

Dataset name	Transaction#	Items#	Size
Chess	3196	175	0.78 M
Foodmart	4141	1559	12.4 M
ChainStore	111,294	46,086	28.17 M



Hình 3.12 – 3.14: Thời gian thực nghiệm trên tập các tập dữ liệu



Hình 3.4: Đánh giá bộ nhớ sử dụng của các thuật toán trên các tập dữ liệu

3.5 Kết luận chương 3

Nhằm tăng tính hiệu quả trong các mô hình dữ liệu lớn, các bản ghi cập nhật liên tục. Chương 3 tập trung trình bày phương pháp khai phá tập mục phổ biến mờ theo kỹ thuật xử lý song song CLA. Theo CLA, không gian được biểu diễn như một mạng, với mỗi phần tử là một ô, từng dòng một, dữ liệu giao dịch sẽ được đọc và đồng thời được chuyển đến các ô, chúng sẽ cộng tác với nhau song song. Với việc không sử dụng quy tắc vùng lân cận, một loại tự động dữ liệu được gọi là tự động học di động bất thường (ICLA) được sử dụng để tạo danh sách vùng lân cận cho mỗi ô. Thông qua việc sử dụng các ô dữ liệu tự động này, việc khai phá tập mờ phổ biến được thực hiện. Quá trình này rút ngắn thời gian thực thi của thuật toán. [CT3].

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Mục đích chính của luận án là nghiên cứu một số phương pháp khai phá luật kết hợp mờ. Luận án nghiên cứu các phương pháp khai phá luật kết hợp trên cơ sở dữ liệu mờ dựa trên sự kết hợp của toán học mờ và cơ sở dữ liệu định lượng đã được đề xuất. Tuy nhiên, các phương pháp này đang trong quá trình phát triển, việc đề xuất các giải pháp mới nhằm hoàn thiện cho các là rất cần thiết. Vì vậy, luận án đề xuất hướng tiếp cận hiệu quả cho vấn đề khai phá các luật kết hợp mờ.

Các kết quả chính của luận án đạt được như sau:

(1) Đề xuất phương pháp xác định các tập mờ cho mỗi thuộc tính định lượng trong cơ sở dữ liệu thông qua kỹ thuật phân cụm EMC. Sau đó, các cụm này được sử dụng để phân loại mỗi thuộc tính định lượng như một tập mờ và xác định các hàm thuộc của chúng. Kết quả của bước này để chuyển đổi cơ sở dữ liệu định lượng sang cơ sở dữ liệu mờ. [CT2], [CT4].

(2) Đề xuất phương pháp khai phá tập mục mờ phổ biến dựa trên cấu trúc Nodelist, bước quan trọng trong khai phá luật kết hợp mờ. Quy trình khai phá tập mục mờ phổ biến dựa trên PP_code hoặc POS_code giúp hạn chế mức tiêu thụ bộ nhớ được yêu cầu. [CT1], [CT2]

(3) Đề xuất phương pháp xử lý song song cho quá trình khai phá tập mục mờ phổ biến bằng cách sử dụng lý thuyết tự động học di động CLA. Với đề xuất này nhằm giải quyết giảm thời gian xử lý cho các cơ sở dữ liệu lớn. [CT3].

DANH MỤC CÔNG TRÌNH CỦA TÁC GIẢ

STT	TÊN BÀI BÁO
[CT1]	Tran, T. T., Nguyen, G. L., Truong, C. N., & Nguyen, T. T. "Mining Frequent Fuzzy Itemsets Using Node-List". Information Systems Design and Intelligent Applications Springer, Singapore, 37-48, 2018
[CT2]	Tran, T. T., Nguyen, T. N., Nguyen, T. T., Nguyen, G. L., & Truong, C. N., "A Fuzzy Association Rules Mining Algorithm with Fuzzy Partitioning Optimization for Intelligent Decision Systems". International Journal of Fuzzy Systems, 1-14, 2022 (SCIE – Q2)
[CT3]	Tran, T. T., Nguyen, T. T., Nguyen, G. L., & Truong, C. N. "Parallel Fuzzy Frequen Itemset Mining Using Cellular Automata". Journal of Computer Science and Cybernetics, 38(4), 293-310, 2022.
[CT4]	Trần Thị Thúy Trinh, Nguyễn Long Giang, Trương Ngọc Châu, Nguyễn Tấn Thuận. "Phân vùng dữ liệu mờ bằng phương pháp thống kê trong khai phá luật kết hợp mờ". Kỷ yếu hội thảo quốc gia về Các vấn đề chọn lọc Công nghệ thông tin và truyền thông – Quy Nhơn), 2017
[CT5]	Trần Thị Thúy Trinh, Nguyễn Tấn Thuận, Nguyễn Long Giang, Trương Ngọc Châu, Nguyễn Quang Huy. "Mô hình tư vấn học tập thông minh ứng dụng luật kết hợp mờ". Hội thảo quốc gia lần thứ XXIII: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Quảng Ninh, 2020